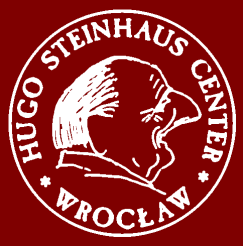


HSC Research Report

HSC/18/05

# Probabilistic electricity price forecasting with NARX networks: Combine point or probabilistic forecasts?

Grzegorz Marcjasz[1,2]
Bartosz Uniejewski[1,2]
Rafał Weron[2]

[1] Faculty of Pure and Applied Mathematics, Wrocław University of Technology, Poland
[2] Department of Operations Research, Faculty of Computer Science and Management, Wrocław University of Technology, Poland

Hugo Steinhaus Center
Wrocław University of Technology
Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
http://www.im.pwr.wroc.pl/~hugo/

# Probabilistic electricity price forecasting with NARX networks: Combine point or probabilistic forecasts?

Grzegorz Marcjasz[a,b], Bartosz Uniejewski[a,b], Rafał Weron[a]

[a] *Department of Operations Research, Faculty of Computer Science and Management, Wrocław University of Science and Technology, Wrocław, Poland*
[b] *Faculty of Pure and Applied Mathematics, Wrocław University of Science and Technology, Wrocław, Poland*

**Abstract**

A recent *electricity price forecasting* (EPF) study has shown that the *Seasonal Component Artificial Neural Network* (SCANN) modeling framework, which consists of decomposing a series of spot prices into a trend-seasonal and a stochastic component, modeling them independently and then combining their forecasts, can yield more accurate point predictions than an approach in which the same non-linear autoregressive NARX-type neural network is calibrated to the prices themselves. Here, considering two novel extensions of the SCANN concept to probabilistic forecasting, we find that (i) efficiently calibrated NARX networks can outperform their autoregressive counterparts, even without combining forecasts from many runs, and that (ii) in terms of accuracy it is better to construct probabilistic forecasts directly from point predictions, however, if speed is a critical issue, running quantile regression on combined point forecasts (i.e., committee machines) may be an option worth considering. Moreover, we confirm an earlier observation that averaging probabilities outperforms averaging quantiles when combining predictive distributions in EPF.

*Keywords:* Electricity spot price, Probabilistic forecast, Combining forecasts, Long-term seasonal component, NARX neural network, Quantile regression

## 1. Introduction

Recent interest in deseasonalizing electricity prices with respect to the *long-term seasonal component* (LTSC) prior to making short-term forecasts has been spurred by the paper of Nowotarski and Weron (2016) and their *Seasonal Component AutoRegressive* (SCAR) modeling framework. However, the idea behind it is not entirely new. In the energy economics literature, Janczura et al. (2013), Keles et al. (2016) and Lisi and Nan (2014), among others, have emphasized that a key point in electricity price modeling is the treatment of seasonality and appropriate seasonal decomposition. Also in the machine learning community, Andrawis et al. (2011) and Zhang and Qi (2005) have argued that neural networks are not able to capture seasonal or trend variations effectively when calibrated to raw data and that detrending and/or deseasonalization can

*Email addresses:* gelusz@hotmail.co.uk (Grzegorz Marcjasz), uniejewskibartosz@gmail.com (Bartosz Uniejewski), rafal.weron@pwr.edu.pl (Rafał Weron)

dramatically reduce forecasting errors. Somewhat surprisingly, though, these facts have not been acknowledged by energy forecasters and in most studies only short-term periodicities have been accounted for (see Weron, 2014, for a review).

In a follow-up study on the importance of trend-seasonal components in day-ahead EPF, Marcjasz et al. (2018) have considered non-linear autoregressive (NARX-type, i.e., with an exogenous variable) neural networks with the same inputs as the SCARX models of Nowotarski and Weron (2016); again 'X' denotes that exogenous variables are utilized. They have shown that while individual *Seasonal Component Artificial Neural Network* (SCANN) models implemented in Matlab are generally worse than the corresponding SCAR-type structures, committee machines of SCANN networks (i.e., combined point forecasts) can outperform the latter significantly. Moreover, that the accuracy gains from using the seasonal component approach are even higher for NARX networks than for their linear counterparts.

In a parallel study, Uniejewski et al. (2018) extended the SCAR concept to probabilistic forecasts, by considering SCARX models with different LTSCs and pooling the resulting prediction errors (via historical simulation or bootstrapping) or the point forecasts themselves (via *Quantile Regression Averaging*, QRA); note, that the latter is similar in spirit to combining so-called *sister forecasts* in load forecasting (see Liu et al., 2017). While the same extension could be considered for NARX-based models, neural networks offer a different, yet potentially even more attractive approach. Namely, due to the calibration algorithm, which is initialized using a random starting point that is independent for each run, each time we estimate weights of a neural net we obtain different values and hence different point forecasts.[1] The latter can be combined in a point forecasting setting and used to yield predictive distributions via quantile regression. Alternatively, the individual point forecasts can be directly combined in a probabilistic setting using QRA (see Nowotarski and Weron, 2015). But which approach is better? And is any of the two worth recommending at all?

With this paper we want address these questions in a comprehensive empirical study that involves two hourly resolution datasets from two distinct power markets (GEFCom2014 and Nord Pool), offering a test ground of nearly six/five years of hourly electricity prices for evaluating point/probabilistic forecasts. We consider four classes of point forecasting models:

1. A naïve similar-day type benchmark.
2. A set of 18 + 1 *Seasonal Component AutoRegressive* (SCAR) models used by Nowotarski and Weron (2016) and Uniejewski et al. (2018); the '+1' refers to a model without the LTSC. All SCAR-type models are built on a popular autoregressive model structure, originally proposed by Misiorek et al. (2006) and later used in a number of EPF studies, after Uniejewski et al. (2016) and Ziel (2016) called an *expert* model.
3. A set of 18 + 1 *Seasonal Component Artificial Neural Network* (SCANN) models used by Marcjasz et al. (2018); like above, the '+1' refers to a model without the LTSC. All SCANN-type models are built on an artificial neural network (ANN) with the same input variables as the expert model, also known as a *non-linear autoregressive* (NAR) model.
4. Committee machines of 2 to 5 SCANN-type networks with identical structures but different weights (also see the discussion in Section 3.1.3).

[1] In contrast, the *ordinary least squares* (OLS) technique always yields the same parameter estimates for an ARX-type model.

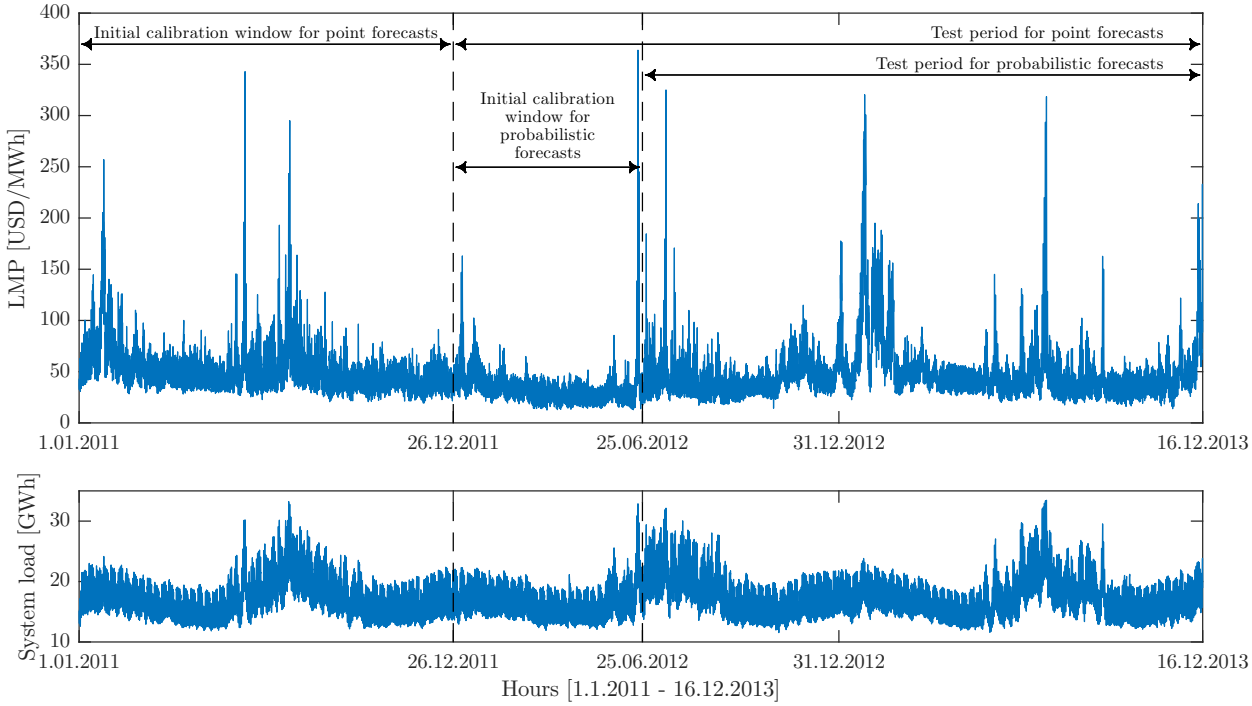


Figure 1: GEFCom2014 hourly locational marginal prices (LMP; *top*) and hourly day-ahead predictions of system load (*bottom*) for the period 1.1.2011–16.12.2013. Probabilistic forecasts are obtained for the 77-week (i.e., 539-day) test period from 26.06.2012 to 16.12.2013.

The obtained point forecasts are used to construct probabilistic predictions via: historical simulation (as a benchmark), *Quantile Regression Averaging* (QRA) of Nowotarski and Weron (2015) and a new approach we call *Quantile Regression Machine* (QRM), which applies quantile regression (see, e.g., Koenker, 2005) to outputs of a committee machine. Furthermore, given a set of probabilistic forecasts we can combine them in one of two ways: by averaging either probabilities or quantiles, see Lichtendahl et al. (2013) for a general discussion and Uniejewski et al. (2018) for an EPF application. We consider both approaches.

The remainder of the paper is structured as follows. In Section 2 we briefly present the datasets, then in Section 3 describe the techniques considered for point and probabilistic EPF. In Section 4 we first summarize the empirical findings in terms of the robust *weekly-weighted mean absolute error* (WMAE; see Weron, 2014) for point forecasts and the *pinball loss* function for probabilistic forecasts (Gneiting, 2011; Hong et al., 2016; Nowotarski and Weron, 2018). Then we report the results of the Diebold and Mariano (1995) test for significant differences in forecasting performance. Finally, in Section 5 wrap up the results and conclude.

## 2. Datasets

We consider two datasets comprising day-ahead time series from two distinct power markets. The first one comes from the Global Energy Forecasting Competition 2014 (GEFCom2014; for details see Hong et al., 2016), and includes three preprocessed (to account for missing values and

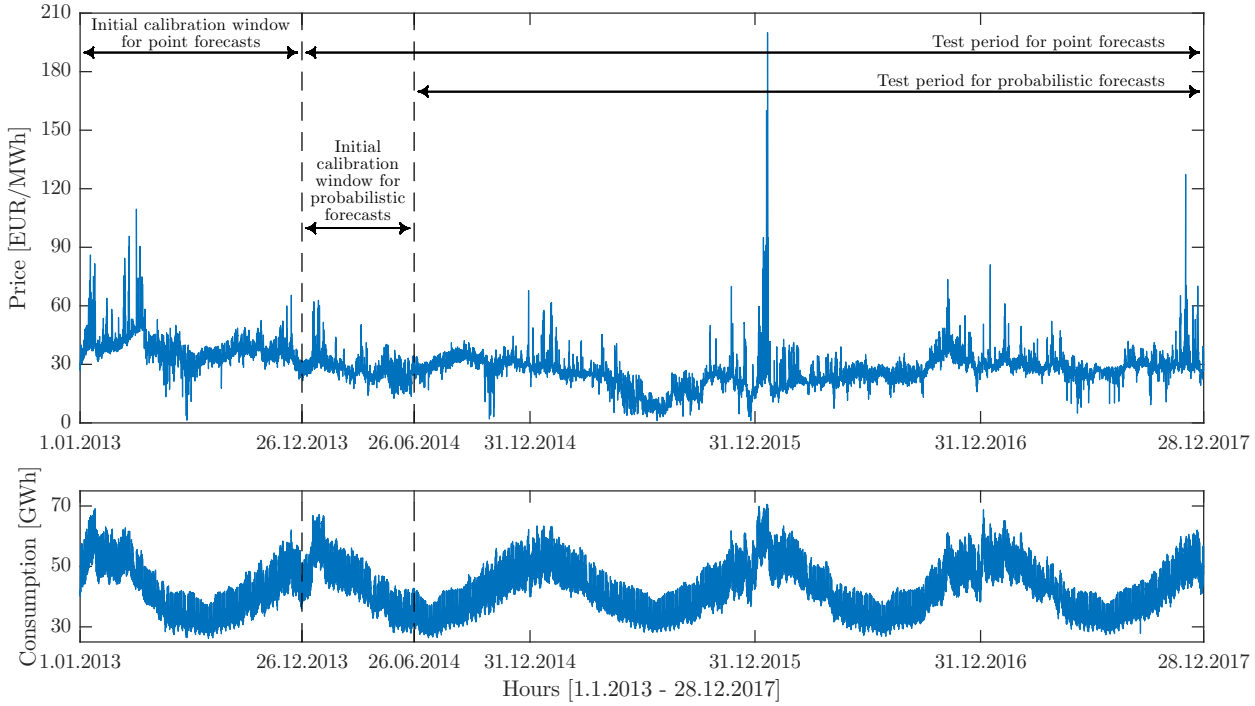


Figure 2: Nord Pool hourly system prices (*top*) and hourly consumption prognosis (*bottom*) for the period 1.1.2013–28.12.2017. Probabilistic forecasts are obtained for the 183-week (i.e., 1281-day) test period from 27.06.2014 to 28.12.2017.

changes to/from the daylight saving time) time series from the period 1.1.2011–16.12.2013: hourly locational marginal prices (LMP) and day-ahead predictions of hourly zonal and system loads. Although the origin of the data has never been revealed by the organizers, the holiday structure suggests that it comes from the US. Like Marcjasz et al. (2018) and Uniejewski et al. (2018), we use LMPs and day-ahead predictions of system loads, see Fig. 1. Note, that Nowotarski and Weron (2016) used zonal loads, but since system loads yield slightly better price predictions, we use them instead.

The second dataset describes one of the major European power markets – Nord Pool (NP) – and comprises hourly system prices and hourly *consumption prognosis* for four Nordic countries (Denmark, Finland, Norway and Sweden) for the period 1.01.2013–28.12.2017, see Fig. 2. The time series was constructed using data published by the Nordic power exchange Nord Pool (www.nordpoolspot.com) and preprocessed to account for missing values and changes to/from the daylight saving time, i.e., the missing data values were substituted by the arithmetic average of the neighboring values, while the 'doubled' values (corresponding to the changes from the daylight saving/summer time) were substituted by the arithmetic average of the two values for the 'doubled' hour.

## 3. Methodology

Like in many EPF studies, the modeling is implemented separately across the hours, leading to 24 sets of parameters for each day the forecasting exercise is performed. This 'multivariate' framework explicitly uses a 'day × hour', matrix-like structure with $P_{d,h}$ representing the price for day $d$ and hour $h$, and implicitly assumes that the error variance is different for each of the 24 load periods (see Ziel and Weron, 2018, for a discussion of the 'uni-' and 'multivariate' frameworks). The 24 individual models are estimated independently, while prices for all load periods of the next day are predicted at once as one-day ahead forecasts.

The point forecasts of the hourly electricity price (see Section 3.1) are determined within a rolling window scheme, using data from the most recent 360 days. Initially all considered models (their short-term and long-term components) are calibrated to data from 1.01.2011 to 26.12.2011 (for GEFCom2014) or from 1.01.2013 to 26.12.2013 (for Nord Pool), and forecasts for all 24 hours of 27th December are determined. Then the window is rolled forward by one day and forecasts for all 24 hours of 28th December are computed. This procedure is repeated until the predictions for the last day in the sample – 16.12.2013 (for GEFCom2014) or 28.12.2017 (for Nord Pool) – are made.

Once the point predictions are made, they are used to provide probabilistic forecasts. All three considered approaches (historical simulation, QRA and QRM; see Section 3.2) require a subsample of one-day ahead prediction errors. Hereby, like in Uniejewski et al. (2018), a 182-day (or 26-week) rolling calibration window is used for computing quantiles of the error distribution (historical *prediction intervals*, PIs) or weights of the QRA/QRM approaches. As a result, probabilistic forecasts are obtained for the periods: 26.06.2012–16.12.2013 (GEFCom2014; 77 full weeks) and 27.06.2014–28.12.2017 (Nord Pool; 183 full weeks), see Figs. 1 and 2.

### *3.1. Point forecasts*

Apart from the more sophisticated SCAR- and SCANN-type models discussed below, we use an extremely simple point forecasting benchmark. The so-called *naïve model* of Nogales et al. (2002) belongs to the class of similar-day techniques and proceeds as follows: the price forecast for hour $h$ on Monday is equal to the price for hour $h$ on Monday of the previous week, i.e., $\hat{P}_{d,h} = P_{d-7,h}$, and the same rule applies for Saturdays and Sundays. For the remaining days, the price forecast for hour $h$ on day $d$ is equal to the price for hour $h$ on day $d-1$, i.e., $\hat{P}_{d,h} = P_{d-1,h}$. Obviously, the benchmark does not require a long calibration window nor parameter estimation. Later in text we denote it by **Naïve**.

#### *3.1.1. SCAR-type models*

The basic building block of the SCAR-type models considered here is a parsimonious autoregressive structure originally proposed by Misiorek et al. (2006) and later used in a number of EPF studies (Gaillard et al., 2016; Kristiansen, 2012; Maciejowska et al., 2016; Nowotarski and Weron, 2016, 2018; Nowotarski et al., 2014; Serinaldi, 2011; Weron, 2006; Weron and Misiorek, 2008; Ziel and Weron, 2018). Following Uniejewski et al. (2016) and Ziel (2016) we refer to it as an *expert* model, since it is built on some prior knowledge of experts. Within this model the

natural logarithm of the electricity spot price is given by: $p_{d,h} \equiv \log(P_{d,h}) = \overline{p} + q_{d,h}$, i.e., the sum of the mean log-price in the calibration window, $\overline{p}$, and an autoregressive component:

$$q_{d,h} = \underbrace{\beta_{h,1} q_{d-1,h} + \beta_{h,2} q_{d-2,h} + \beta_{h,3} q_{d-7,h}}_{\text{autoregressive effects}} + \underbrace{\beta_{h,4} q_{d-1,\min}}_{\text{non-linear effect}} + \underbrace{\beta_{h,5} z_t}_{\text{load}} + \underbrace{\sum_{i=1}^{3} \beta_{h,i+5} D_i}_{\text{weekday dummies}} + \varepsilon_{d,h}, \qquad (1)$$

where $q_{d-1,\min} = \min_{h=1,...,24}\{q_{d-1,h}\}$ is the minimum of the previous day's 24 hourly prices and creates a link with all yesterday's prices, not just the prices for the same hour. The variable $z_t$ is the logarithm of the day-ahead forecasts of either the hourly system load of a US utility or of the Nordic consumption. The three dummy variables – $D_1$, $D_2$ and $D_3$ (for Monday, Saturday and Sunday, respectively) – account for the weekly seasonality. Finally, the $\varepsilon_{d,h}$'s are assumed to be independent and identically distributed (i.i.d.) normal variables. To reflect the fact that $z_t$ is an eXogenous variable in Eqn. (1), we denote this basic model by **ARX**.

The *Seasonal Component AutoRegressive* (SCAR) modeling framework of Nowotarski and Weron (2016) is motivated by the standard approach to seasonal decomposition, where a time series is decomposed into the long- and short-term seasonal components, and the remaining variability or stochastic component (Hyndman and Athanasopoulos, 2013; Weron, 2014). More precisely, in the SCAR framework the electricity log-price is decomposed in an additive manner into a LTSC and a stochastic component with short-term periodicities: $p_{d,h} = T_{d,h} + q_{d,h}$. Note, that compared to the original SCAR algorithm, following Uniejewski et al. (2018) we add here an additional step, called 1(b), which significantly improves the forecasting performance. The full SCAR algorithm consists of the following four steps:

1. (a) Decompose the series of log-prices $p_{d,h}$ in the calibration window of $360 \times 24 = 8640$ hours into a long-term seasonal component $T_{d,h}$ and a stochastic component with short-term periodicities $q_{d,h}$. Then compute persistent forecasts of the LTSC independently for each of the 24 hours of the next day, i.e., $\hat{T}_{d^*+1,h} \equiv T_{d^*,h}$ for $h = 1, ..., 24$, where $d^*$ is the last day in the calibration window, see Fig. 2 in Marcjasz et al. (2018) for an illustration.
   (b) Decompose the exogenous series (the logarithm of the system load or consumption forecast) in the calibration window of $(360 + 1) \times 24 = 8664$ hours using the same type of a LTSC as prices in Step 1(a). Note, that we can start one day earlier since the load or consumption forecasts are known one day in advance.
2. Calibrate the **ARX** model defined by Eqn. (1) to $q_{d,h}$ and compute forecasts for the 24 hours of the next day, i.e., $\hat{q}_{d^*+1,h}$. Note, that unlike the seasonal decomposition in Step 1, which is made for the whole calibration sample, here the data is split into 24 hourly series (like for the **ARX** benchmark).
3. Add forecasts of the **ARX** model computed in Step 2 to the persistent forecasts of the LTSC to yield log-price forecasts: $\hat{p}_{d^*+1,h}$.
4. Take the exponent of the log-price forecasts computed in Step 3 to convert them into price forecasts of the **SCARX** model: $\hat{P}_{d,h} = \exp(\hat{p}_{d,h})$.

Like Nowotarski and Weron (2016), Marcjasz et al. (2018) and Uniejewski et al. (2018), in what follows we consider 18 + 1 **SCARX** models which differ in the choice of the LTSC; the

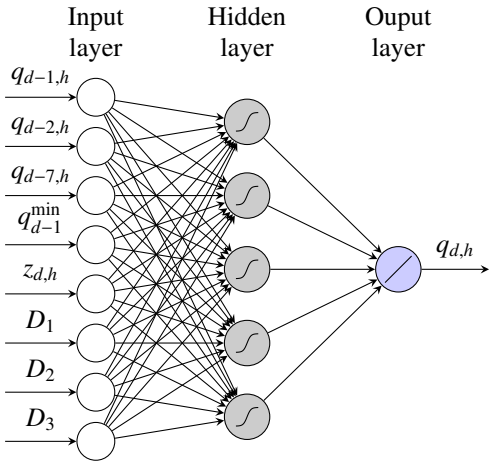


Figure 3: Visualization of the NARX network with the same inputs as the **ARX** model defined by Eqn. (1), five hidden neurons with tangent sigmoid transfer functions and one output neuron with a linear transfer function.

'+1' refers to a model without the LTSC, i.e., the **ARX** model itself. For the LTSC we either use one of ten wavelet filters ($S_5, S_6, ..., S_{14}$ based on the Daubechies family of order 24; see Afanasyev and Fedorova, 2016; Janczura et al., 2013; Nowotarski et al., 2013, for details) or one of eight Hodrick-Prescott filters (HP with $\lambda = 10^8, 5 \cdot 10^8, ..., 5 \cdot 10^{11}$; see Caldana et al., 2017; Lisi and Nan, 2014; Weron and Zator, 2015, for details). This modeling choice is justified by the fact that in most electricity markets the annual weather-driven seasonality is dominated by an irregular cyclic behavior (i.e., not of a fixed period, as opposed to the popular sine/cosine-based seasonal components) reflecting prevailing macroeconomic conditions, long-term weather trends and changes in strategic bidding practices.

*3.1.2. SCANN-type models*

Like **ARX** is the basic building block of the SCAR-type models, a *non-linear autoregressive* model with eXogenous variables (NARX) is the basic building block of the SCANN-type models considered in this study. The NARX network is a recurrent neural network (RNN), with the output being fed back to the input of the network. Since the true output is available during the training of the network, it can be fed back instead of the estimated output, leading to a feed-forward representation (Hagan et al., 2014). The rationale for using this computationally efficient architecture (instead of, e.g., a *long short-term memory* model of Hochreiter and Schmidhuber, 1997) is that only short term dynamics is expected to remain after filtering out the LTSC.

To represent what they called the **ANN** model, Marcjasz et al. (2018) used Matlab's *narxnet* structure with exactly the same inputs as those of the **ARX** model in Eqn. (1), one hidden layer consisting of five neurons (with tangent sigmoid transfer functions) and an output layer with one neuron yielding $q_{d^*+1,h}$ (with a linear transfer function), see Fig. 3. Formally, **ANN** is a misnomer since the load or consumption forecast is used as the eXogenous variable. However, to simplify notation we also do not explicitly use X in the model name.

Marcjasz et al. (2018) trained the **ANN** model in Matlab using the Levenberg-Marquardt algorithm (function *trainlm.m*). Here, to improve computational efficiency we utilize Python's inter-

face to the *Fast Artificial Neural Networks* (FANN) library.[2] We use the same network structure (see Fig. 3), but a different training algorithm – the so-called *incremental* scheme. The training parameters have been selected based on a limited empirical study, and – in order to provide a fair test ground – chosen to be identical for both datasets and across all LTSCs.[3] As far as computational efficiency is concerned, the FANN-trained models not only were much faster (up to 40 times!), but also significantly outperformed Matlab's forecasts in terms of accuracy.

The *Seasonal Component Artificial Neural Network* (SCANN) modeling framework is a generalization of the **ANN** model in the same way the SCAR framework is built on the **ARX** model, i.e., it consists of four steps analogous to those discussed in Section 3.1.1, only with **ARX** replaced by **ANN** (see Marcjasz et al., 2018, for details). Like for the SCAR-type models, in what follows we consider 18 + 1 **SCANN** models which differ in the choice of the LTSC; the '+1' refers to a model without the LTSC, i.e., the **ANN** model itself. For the LTSC we either use one of ten wavelet filters ($S_5, S_6, ..., S_{14}$) or one of eight HP-filters (with $\lambda = 10^8, 5 \cdot 10^8, ..., 5 \cdot 10^{11}$).

### *3.1.3. Committee machines*

The NARX calibration algorithm is initialized using a random starting point that is independent for each run (and each day and hour), and hence yields different estimates for each run. Quite often the resulting variance of the forecasts (across the runs) is not negligible. Therefore a reasonable approach may be to repeat every training and forecasting exercise $n$ times and average the point forecasts on an hour-by-hour basis across the runs, like in Marcjasz et al. (2018) and Shrivastava and Panigrahi (2014). We denote such models by $\mathbf{SCANN}_n$ (or $\mathbf{ANN}_n$) and refer to them as *committee machines*, *ensemble averages* or *combined forecasts*. On the other hand, by $\overline{\mathbf{SCANN}}_1$ (or $\overline{\mathbf{ANN}}_1$) we denote the expected value of a single (SC)ANN network, computed as the average WMAE (see Section 4.1) across the runs. Although we have considered $n$ as large as 25, we present results for $n = 5$ only. Increasing the size of the ensemble further has a positive but decreasing with $n$ effect on forecast accuracy and is not justified by the substantially heavier computational burden.

## *3.2. Probabilistic forecasts*

The most common extension from point to probabilistic forecasts is to construct *prediction intervals* (PIs). In this study, like in the GEFCom2014 competition, we consider all 99 percentiles (1%, 2%, ..., 99%; not only two subjectively selected quantiles as in may other studies, see Nowotarski and Weron, 2018, for a review). This allows us for a reasonably accurate approximation of the whole predictive distribution, not just of one PI.

We use two well known and one new method of obtaining probabilistic forecasts, see Fig. 4. Note, that each of the probabilistic models is characterized by parameter $N$ indicating the number of independent point forecasts used (i.e., 'runs' of the neural net). We have set the upper limit of $N$ to be 5, since higher values are no longer beneficial for the forecasting potential of committee machines (i.e., the inputs of the **QRM**($N$) model; see Section 3.2.3) and lead to substantially longer

[2] See `http://leenissen.dk/fann/wp/` and Nissen (2007).

[3] *Hidden layer steepness* = 0.6, *Output layer steepness* = 0.3, *Learning rate* = 0.2, *Randomized weights range* = $[-1, 1]$; for the remaining parameters we have used the default values.

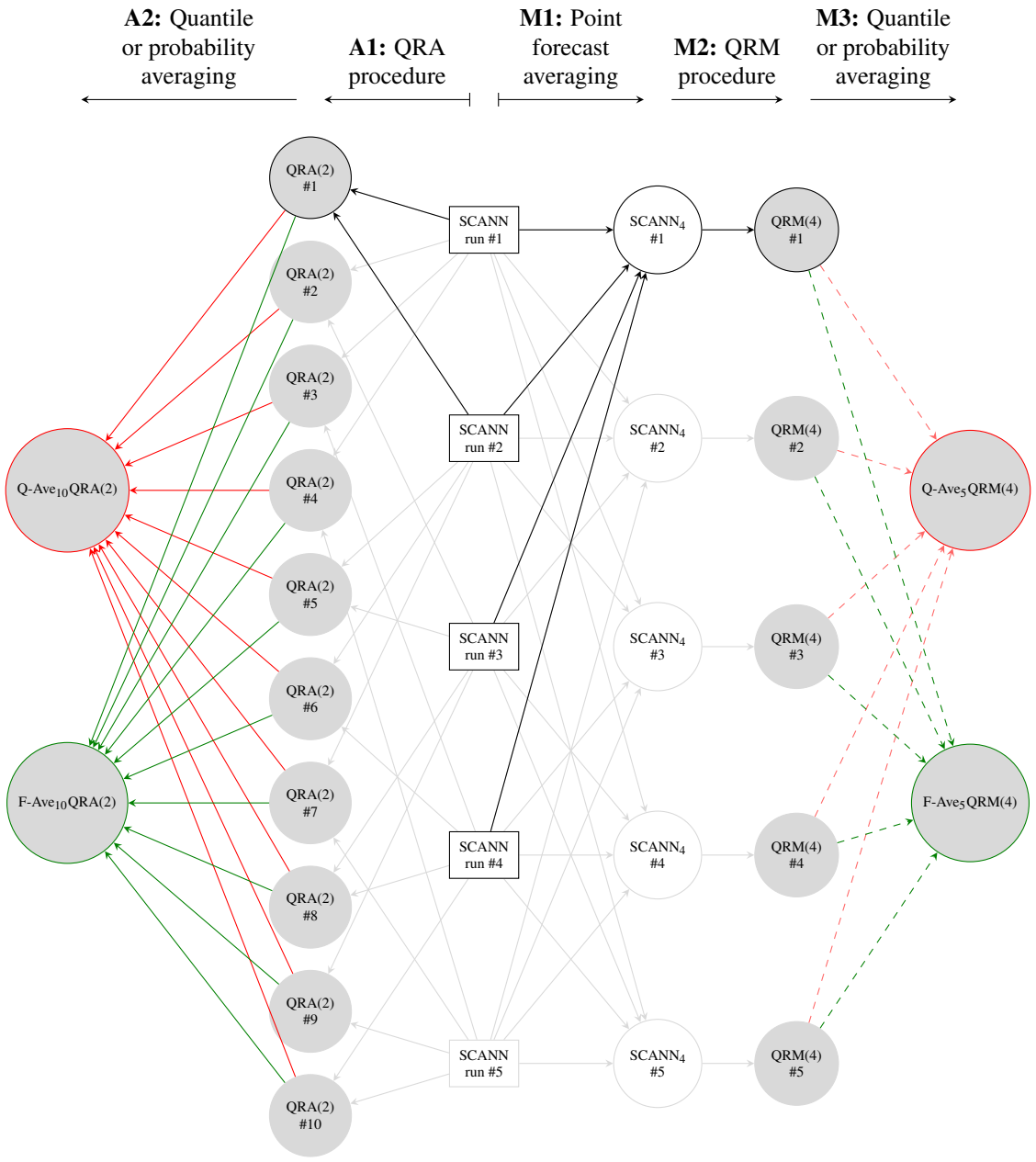


Figure 4: Visualization of the QRA- (steps **A1-A2**) and QRM-based (steps **M1-M3**) concepts of computing probabilistic forecasts. Initial point forecasts (SCANN #1-#5) are depicted by rectangles. Small white/gray circles represent point/probabilistic predictions and large gray circles refer to averages of probabilistic forecasts. Black arrows represent either the process of constructing a single **QRA(2)** forecast from two SCANN predictions (*top left*) or a single **QRM(4)** probabilistic forecast from four SCANN predictions via a committee machine (*top center/right*); gray arrows illustrate the remaining links. Colored arrows refer to quantile (*red*) or probability (*green*) averages constructed from **QRA** (*solid*) or **QRM** (*dashed*); the color/style scheme is the same as in Figs. 5 and 6.

computational times for the quantile regression averaging scheme (i.e., the **QRA**(*N*) models; see Section 3.2.2). Therefore all of the probabilistic results originate from 5 independent SCANN runs.

*3.2.1. Historical simulation*

The first approach is a simple, model-independent technique that consists of computing sample quantiles of the empirical distribution $\hat{F}_{\varepsilon_{d,h}}$ of day-ahead prediction errors $\varepsilon_{d,h}$ (= $\hat{P}_{d,h} - P_{d,h}$) centered around the point forecast $\hat{P}_{d^*+1,h}$.[4] Recall, that every run of the SCANN network yields errors for the whole length of the test period, i.e., $\{\varepsilon_{1,1}, ..., \varepsilon_{\mathcal{D},24}\}$ with $\mathcal{D} = 539$ for GEFCom2014 and 1281 for Nord Pool. However, to improve accuracy we can pool $\varepsilon_{d,h}$'s from $N$ runs, i.e., $\{\varepsilon^1_{1,1}, ..., \varepsilon^1_{\mathcal{D},24}, ..., \varepsilon^N_{1,1}, ..., \varepsilon^N_{\mathcal{D},24}\}$, and compute $\hat{F}_{\varepsilon_{d,h}}$ implied by this joint vector; the resulting approach is denoted by **Hist**(*N*). We also consider historical simulation for single runs, i.e., **Hist**(1), and denote its expected value by $\overline{\textbf{Hist}(1)}$; the latter is computed as the average pinball score (see Section 4.2) across the runs. This concept is more general and can be extended to pairs of runs, i.e., **Hist**(2) with expected value $\overline{\textbf{Hist}(2)}$, triples and quadruples.

Although historical simulation is not visualized explicitly in Fig. 4, the idea behind it can be explained using this diagram. For instance, **Hist**(4) can be constructed by taking the #1 $\textbf{SCANN}_4$ committee machine (*top center/right*) as the point forecast and the empirical distribution $\hat{F}_{\varepsilon_{d,h}}$ implied by day-ahead prediction errors for runs #1-#4 of the SCANN network (*center column*), while **Hist**(1) is computed by taking the point forecast and $\hat{F}_{\varepsilon_{d,h}}$ implied by errors of a single run of the SCANN network. $\overline{\textbf{Hist}(1)}$ can then be approximated by the average across runs #1-#5.

*3.2.2. Quantile Regression Averaging*

This technique, originally proposed by Nowotarski and Weron (2015), has been found to perform very well in a number of test cases (see, e.g., Gaillard et al., 2016; Maciejowska and Nowotarski, 2016; Maciejowska et al., 2016), not only in the area of EPF (Liu et al., 2017; Zhang et al., 2016). However, its most spectacular success came during the GEFCom2014 competition – the top two winning teams in the price track used variants of QRA (Gaillard et al., 2016; Maciejowska and Nowotarski, 2016). Quantile Regression Averaging (QRA) involves applying quantile regression (see, e.g., Koenker, 2005) to a pool of point forecasts of individual (i.e., not combined) forecasting models. As such, it directly works with the distribution of the electricity spot price, $\hat{F}_{P_{d,h}}$, without the need to split the probabilistic forecast into a point forecast $\hat{P}_{d^*+1,h}$ and the distribution of the error term $\hat{F}_{\varepsilon_{d,h}}$. Later in the text we denote this method by **QRA**(*N*), where *N* refers to the number of individual point forecasts from which the probabilistic forecast is computed.

The QRA approach is visualized in the left part of Fig. 4 as step **A1**. In particular, probabilistic forecasts of the #1 **QRA**(2) model (*top left*) are obtained by applying quantile regression to the point forecasts of runs #1 and #2 of the SCANN network in the 182-day 'probabilistic' calibration window, see Figs. 1-2. There are 10 different **QRA**(2) models in the diagram, because there are $\binom{5}{2} = 10$ ways of selecting two runs out of five. Similarly, there would be five different **QRA**(1) and **QRA**(4) models, 10 different **QRA**(3) models and only one **QRA**(5) model. As discussed in Section 3.2.4, the **QRA**(*N*) probabilistic forecasts can be further combined using

[4] As in Section 3.1.1, $d^*$ denotes here the last day in the calibration window.

quantile or probability averaging, as are #1-#10 **QRA**(2) distributional predictions in step **A2** of Fig. 4. Moreover, analogously to historical simulation, we can consider the expected value $\overline{\textbf{QRA}(N)}$ of **QRA**($N$) forecasts, computed as the average pinball score (see Section 4.2) of the latter.

*3.2.3. Quantile Regression Machine*

The last approach can be treated as a variant of QRA or as a new averaging method. Namely, *Quantile Regression Machine*, denoted by **QRM**($N$), applies quantile regression to the point forecasts of a single committee machine, which in turn is based on $N$ runs of a SCANN network. Like QRA it directly works with the distribution of the electricity spot price and uses the same pool of individual point forecasts, but unlike QRA it uses only one (averaged) electricity spot price prediction for each point in time $(d, h)$. Moreover, analogously to QRA, we can consider the expected value $\overline{\textbf{QRM}(N)}$ of **QRM**($N$) forecasts, computed as the average pinball score (see Section 4.2) of the latter.

The QRM approach is visualized in the right part of Fig. 4. In particular, probabilistic forecasts of the #1 **QRM**(4) model (*top right*) are obtained in step **M2** by applying quantile regression to the point forecasts of the #1 $\textbf{SCANN}_4$ committee machine; the latter is computed earlier in step **M1** as an average of the price predictions from runs #1 to #4 of the SCANN network. There are five different **QRM**(4) models in the diagram (corresponding to five $\textbf{SCANN}_4$ committee machines), because there are $\binom{5}{4} = 5$ ways of selecting four runs out of five. Like QRA predictions, the **QRM**(4) probabilistic forecasts can be further combined using quantile or probability averaging (*center right*, i.e., step **M3**).

*3.2.4. Combining probabilistic forecasts*

As Lichtendahl et al. (2013) argue, given a set of $n$ probabilistic forecasts we can combine them in one of two ways: by averaging either probabilities or quantiles. The average probability forecast $\textbf{F-Ave}_n \equiv \frac{1}{n}\sum_{i=1}^{n}\hat{F}_i(x)$, where $\hat{F}_i(x)$ is the $i$-th distributional forecast, can be regarded as a vertical average of the corresponding predictive distributions. On the other hand, the average quantile forecast $\textbf{Q-Ave}_n \equiv \hat{Q}^{-1}(x)$ with $\hat{Q}(x) = \frac{1}{n}\sum_{i=1}^{n}\hat{Q}_i(x)$, where $\hat{Q}_i(x) = \hat{F}_i^{-1}(x)$ is the $i$-th quantile forecast, as a horizontal average. Note that the average quantile forecast is always sharper, i.e., $\textbf{Q-Ave}_n$ has lower variance than $\textbf{F-Ave}_n$. While this feature is an advantage in many forecasting problems (Lichtendahl et al., 2013), in EPF it may not necessarily be so (Uniejewski et al., 2018).

We use suffix **-Hist**($N$) to denote combined probabilistic forecasts obtained via historical simulation, **-QRA**($N$) to refer to QRA-implied forecasts and **-QRM**($N$) for quantile regression forecasts obtained using committee machine predictors. The two combination schemes are visualized in Fig. 4 (*center left* and *center right*). For instance, $\textbf{F-Ave}_{10}$**-QRA**(2) is obtained in step **A2** as a vertical average of 10 predictive distributions, i.e., #1-#10 **QRA**(2), while $\textbf{Q-Ave}_5$**-QRM**(4) is obtained in step **M3** as a horizontal average of five predictive distributions, i.e., #1-#5 **QRM**(4).

*3.2.5. Notes on implementation and computational efficiency*

We should note here, that we utilize a slightly different implementation of quantile regression than in Uniejewski et al. (2018). Due to convergence issues observed for some of our test cases when running Matlab's Nedler-Mead simplex algorithm, we have decided to use Python's *SciPy*

package instead (Jones et al., 2001). Similarly as for the FANN package, there were numerous options to choose from, and once again, the final method (the TNC algorithm) was selected based on a limited simulation study. To ensure a reasonable balance between computational time and accuracy, we make the number of iterations dependent on the size of the problem (i.e., the number of point predictions used as inputs to the QRA algorithm): the larger the problem, the more iterations are allowed. Such a setting is the default one in both Matlab and Python.

It is also important to comment on the computational time needed to obtain the probabilistic forecasts. For the sake of comparison, we have collected total times needed to obtain one week of hourly predictions on a machine equipped with an i5-3570 processor and utilizing all four cores/threads:

- ca. 0.23 of a second for a single run of the SCANN network,
- ca. 25 seconds for generating **QRM(*N*)** forecasts; note, that the computational time does not depend on $N$, because the input to the QRM method always consists of a single point forecast (of a committee machine),
- ca. $68, 111.5, 171.5, 248$ seconds respectively for $N = 2, 3, 4, 5$; the growth is steeper than a linear function of $N$, because – apart from the greater complexity of the problem itself – the algorithm needed more iterations to converge for larger $N$,
- for combining probabilistic forecasts the computational time is negligible, i.e., less than 0.01 of a second.

The main message from this comparison is that QRM may be better suited for time constrained (e.g., real-time) probabilistic forecasting tasks than QRA, despite a slightly worse performance for the same number of SCANN point forecasts; see Sections 4.2-4.3 for details.

## 4. Empirical results

In this Section we present the day-ahead forecasting results. For point forecasts we use a two- (GEFCom2014) / four-year (Nord Pool) out-of-sample test period, for probabilistic – a 182-day shorter test period (for the reasons discussed in Section 3). Recall that models are re-estimated on a daily basis. Price forecasts $\hat{P}_{d^*+1,1}, ..., \hat{P}_{d^*+1,24}$ for all 24 hours of the next day are determined at the same point in time and the 360-day calibration window is rolled forward by one day: $d^* \rightarrow d^* + 1$.

### *4.1. WMAE and the evaluation of point forecasts*

Following Conejo et al. (2005), Weron and Misiorek (2008) and Nowotarski et al. (2014), we compare the models in terms of the *Weekly-weighted Mean Absolute Error* (WMAE) loss function. WMAE is a robust measure similar to MAPE but with the absolute error normalized by the mean weekly price to avoid the adverse effect of negative and close to zero electricity spot prices. We evaluate the forecast performance using weekly time intervals, each with $24 \times 7 = 168$ hourly observations. Note that we also analyzed the forecasts using squared error losses, however,

Table 1: Average WMAE in percent for all 103 (GEFCom2014; *upper half*) or all 209 weeks (Nord Pool; *lower half*) of the out-of-sample test period. Results for the best performing model in each row are emphasized in bold. Note, that for the GEFCom2014 dataset, the results for the **SCARX** models are the same as in Marcjasz et al. (2018), but different for the (SC)ANN models due to a change of the training algorithm. In particular, now $\overline{\textbf{SCANN}}_1$ models outperform **SCARX** models.

| | | | | | **GEFCom2014** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | *Benchmarks* | | | | | |
| | | | | **Naïve** | **ARX** | $\overline{\textbf{ANN}}_1$ | $\textbf{ANN}_5$ | | | |
| | | | | 14.716 | 11.232 | 10.359 | **10.228** | | | |
| | | | | | *SCARX / SCANN with wavelet approximation of price and load* | | | | | |
| | $S_5$ | $S_6$ | $S_7$ | $S_8$ | $S_9$ | $S_{10}$ | $S_{11}$ | $S_{12}$ | $S_{13}$ | $S_{14}$ |
| **SCARX** | 12.917 | 12.226 | 11.106 | 10.849 | **10.732** | 10.776 | 10.843 | 10.824 | 11.100 | 11.072 |
| $\overline{\textbf{SCANN}}_1$ | 12.791 | 12.222 | 10.768 | 10.312 | 10.071 | **10.067** | 10.193 | 10.197 | 10.347 | 10.370 |
| $\textbf{SCANN}_5$ | 12.769 | 12.197 | 10.728 | 10.259 | 10.002 | **9.972** | 10.074 | 10.081 | 10.220 | 10.237 |
| | | | | | *SCARX / SCANN with HP filter on price and load ($\lambda$)* | | | | | |
| | $10^8$ | $5 \cdot 10^8$ | $10^9$ | $5 \cdot 10^9$ | $10^{10}$ | $5 \cdot 10^{10}$ | $10^{11}$ | $5 \cdot 10^{11}$ | | |
| **SCARX** | 10.519 | 10.447 | **10.437** | 10.495 | 10.559 | 10.798 | 10.897 | 11.060 | | |
| $\overline{\textbf{SCANN}}_1$ | 10.275 | **10.169** | 10.173 | 10.265 | 10.327 | 10.418 | 10.406 | 10.382 | | |
| $\textbf{SCANN}_5$ | 10.214 | 10.095 | **10.094** | 10.168 | 10.232 | 10.295 | 10.275 | 10.244 | | |
| | | | | | **Nord Pool** | | | | | |
| | | | | | *Benchmarks* | | | | | |
| | | | | **Naïve** | **ARX** | $\overline{\textbf{ANN}}_1$ | $\textbf{ANN}_5$ | | | |
| | | | | 9.294 | 8.051 | 8.013 | **7.839** | | | |
| | | | | | *SCARX / SCANN with wavelet approximation of price and load* | | | | | |
| | $S_5$ | $S_6$ | $S_7$ | $S_8$ | $S_9$ | $S_{10}$ | $S_{11}$ | $S_{12}$ | $S_{13}$ | $S_{14}$ |
| **SCARX** | 9.267 | 8.990 | 7.954 | 7.747 | 7.707 | **7.668** | 7.862 | 7.942 | 8.032 | 7.972 |
| $\overline{\textbf{SCANN}}_1$ | 9.202 | 8.846 | 7.758 | **7.433** | 7.468 | 7.515 | 7.725 | 8.044 | 7.913 | 8.014 |
| $\textbf{SCANN}_5$ | 9.168 | 8.801 | 7.697 | **7.353** | 7.389 | 7.422 | 7.623 | 7.907 | 7.776 | 7.874 |
| | | | | | *SCARX / SCANN with HP filter on price and load ($\lambda$)* | | | | | |
| | $10^8$ | $5 \cdot 10^8$ | $10^9$ | $5 \cdot 10^9$ | $10^{10}$ | $5 \cdot 10^{10}$ | $10^{11}$ | $5 \cdot 10^{11}$ | | |
| **SCARX** | **8.007** | 8.032 | 8.051 | 8.081 | 8.075 | 8.078 | 8.111 | 8.269 | | |
| $\overline{\textbf{SCANN}}_1$ | **7.758** | 7.767 | 7.784 | 7.846 | 7.851 | 7.916 | 7.963 | 8.144 | | |
| $\textbf{SCANN}_5$ | **7.687** | 7.688 | 7.701 | 7.751 | 7.750 | 7.807 | 7.849 | 8.010 | | |

results were qualitatively similar and are omitted here due to space limitations. For each week we calculate the WMAE for model $i$ as:

$$\text{WMAE}_i = \frac{1}{\bar{P}_{168}}\text{MAE}_i = \frac{1}{168 \cdot \bar{P}_{168}} \sum_{d=1}^{7} \sum_{h=1}^{24} \left| P_{d,h} - \hat{P}^i_{d,h} \right|, \tag{2}$$

where $P_{d,h}$ is the actual price for day $d$ and hour $h$ (not the log-price $p_{d,h}$), $\hat{P}^i_{d,h}$ is the predicted price for that day and hour obtained from model $i$ and $\bar{P}_{168}$ is the mean price for a given week. Note, that WMAE requires the test period to be a multiple of a week (or 168 hours). Hence, when computing WMAE we consider 103 weeks (27.12.2011–16.12.2013) for the GEFCom2014 dataset and 209 weeks (27.12.2013–28.12.2017) for the Nord Pool dataset, see Figs. 1 and 2.

In Table 1 we report the average WMAE in percent for all considered models: four 'simple' benchmarks – **Naïve**, **ARX**, $\overline{\textbf{ANN}}_1$ and $\textbf{ANN}_5$, and three seasonal component (SC) model classes

– **SCARX**, $\overline{\textbf{SCANN}}_1$ and $\textbf{SCANN}_5$ – computed for ten wavelet-based and eight HP filter-based LTSCs. Several interesting conclusions can be drawn:

- Neural network forecasts outperform **(SC)ARX** counterparts for every LTSC. This is true not only for $\textbf{(SC)ANN}_5$ models but also for $\overline{\textbf{(SC)ANN}}_1$. Apparently, the FANN-trained models are much better than the Matlab-trained nets considered in Marcjasz et al. (2018), at least for these datasets.
- The gains from using committee machines ($\overline{\textbf{SCANN}}_1 \rightarrow \textbf{SCANN}_5$) are rather small, definitely much less pronounced than for the Matlab-trained nets in Marcjasz et al. (2018).
- As opposed to the results in Marcjasz et al. (2018), here wavelet-based LTSCs are uniformly better than HP-based. This difference cannot be attributed to the longer Nord Pool dataset, since it is also visible for the GEFCom2014 test sample (the same in both studies).
- Gains from using the SC framework with NARX networks are highly dependent on the dataset. For GEFCom2014, relative gains from choosing the best performing LTSC are much lower for $\textbf{SCANN}_5$ than for **SCARX** models, i.e., 2.5% vs. 7%. On the other hand, for the Nord Pool dataset the gains are 6.2% and 4.7%, respectively.

Up to this point, we have been discussing results obtained for all 18 considered LTSCs. In the Sections to follow, to simplify the notation and focus on the differences between methods and averaging schemes, we will present results only for models with the $S_9$ wavelet-based seasonal component. Our choice is motivated by the very good performance of this LTSC for both datasets, though not the best for each of them. In fact, we found the best LTSC to be one of the three: $S_8$, $S_9$ and $S_{10}$, with a slightly better performance of $S_{10}$ for GEFCom2014 and $S_8$ for Nord Pool. Therefore, $S_9$ can be considered as a safe choice that performs well for both datasets. This also means, that there is still some potential for fine-tuning, similarly to the choice of the neural network training parameters (see the discussion in Section 3.1.2).

### *4.2. Pinball loss and the evaluation of probabilistic forecasts*

We now turn to probabilistic forecasts and – like in the paper of Uniejewski et al. (2018) on SCAR-type models and the GEFCom2014 competition – measure the *sharpness* or concentration of predictive distributions. Sharpness can be evaluated using so-called *proper* scoring rules, for instance, the *pinball loss* (Gneiting, 2011; Nowotarski and Weron, 2018):

$$\text{Pinball}\left(\hat{Q}_{P_{d,h}}(q), P_{d,h}, q\right) = \begin{cases} (1-q)\left(\hat{Q}_{P_{d,h}}(q) - P_{d,h}\right), & \text{for } P_{d,h} < \hat{Q}_{P_{d,h}}(q), \\ q\left(P_{d,h} - \hat{Q}_{P_{d,h}}(q)\right), & \text{for } P_{d,h} \geq \hat{Q}_{P_{d,h}}(q), \end{cases} \tag{3}$$

where $\hat{Q}_{P_{d,h}}(q)$ is the price forecast at the $q$-th quantile and $P_{d,h}$ is the actually observed price for day $d$ and hour $h$. To provide an aggregate score we average Pinball$(\cdot, \cdot, q)$ across all hours in the test period and across all 99 percentiles ($q = 0.01, 0.02, ..., 0.99$; as in the GEFCom2014 competition, see Hong et al., 2016). Naturally, a lower score indicates a better probabilistic forecast, i.e., a more concentrated predictive distribution.

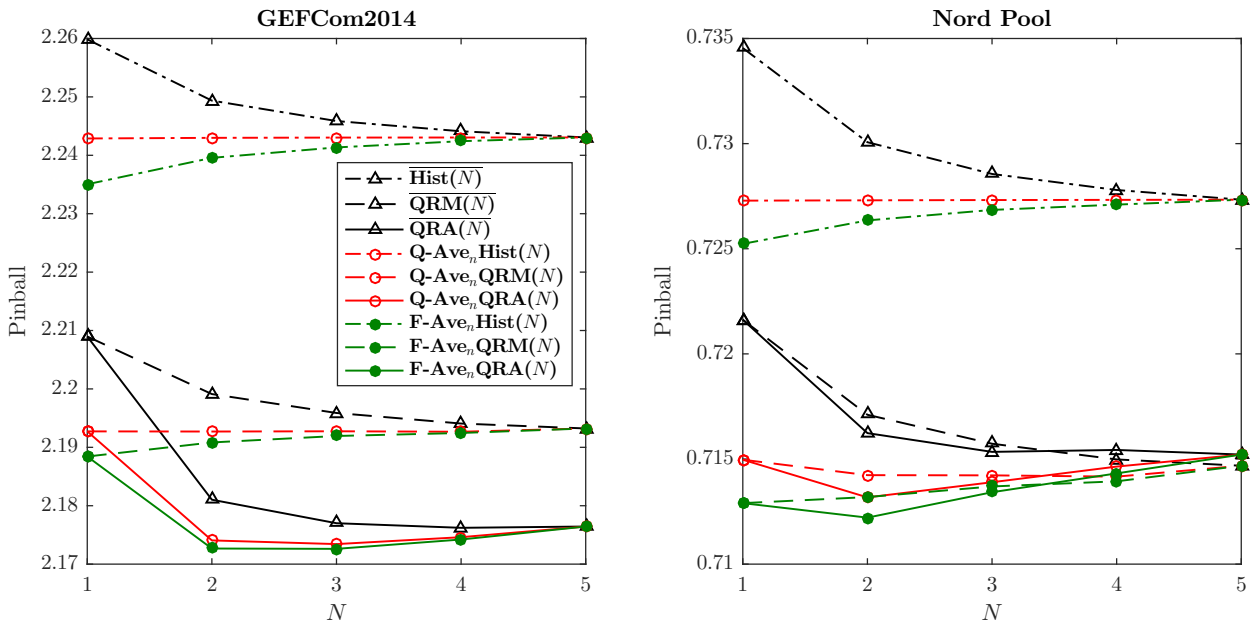


Figure 5: The pinball loss defined by Eqn. (3) averaged across all 99 percentiles and all hours in the 'probabilistic' test period: 77 weeks for GEFCom2014 (*left panel*) and 183 weeks for Nord Pool (*right panel*). Each probabilistic forecast is based on the same five point forecasts of the SCANN network. Black lines can be interpreted as expected values of **Hist**($N$), **QRM**($N$) or **QRA**($N$), red lines refer to average quantile forecasts (**Q-Ave**$_n*$), while green lines represent average probability forecasts (**F-Ave**$_n*$).

In Figure 5 we compare the three methods for computing probabilistic forecasts discussed in Sections 3.2.1-3.2.3, with or without **F-Ave** and **Q-Ave** averaging schemes, as a function of the number $N$ of point forecasts being used. We can observe that:

- For both datasets, historical simulation (dash-dotted lines) is significantly outperformed by QRM (dashed lines), which in turn is nearly in all cases outperformed by QRA (solid lines). The latter is clearly visible for GEFCom2014, but for Nord Pool data the QRA and QRM curves almost overlap. Nevertheless, to answer the question posed in the title, we can say that it is better to combine probabilistic than point forecasts.

- The QRA curves are convex functions with a minimum at $N = 2$ (or 3) for GEFCom2014 and at $N = 2$ for Nord Pool. Since they lie lower than the corresponding QRM curves, we can conclude that **QRA**(2) is the best performer overall.

- The results obtained for QRA with $N > 3$ are clearly worse, and that is, to the best of our knowledge, a result of increasing the numerical complexity of the problem.

- Regarding the averaging scheme – combining predictive distributions is always beneficial, and averaging probabilities (**F-Ave**) outperforms averaging quantiles (**Q-Ave**) in every case.

As was mentioned in Section 3.2.5, the time required to compute an additional point forecast is negligible compared to computing a well performing probabilistic forecast. Hence, it may be

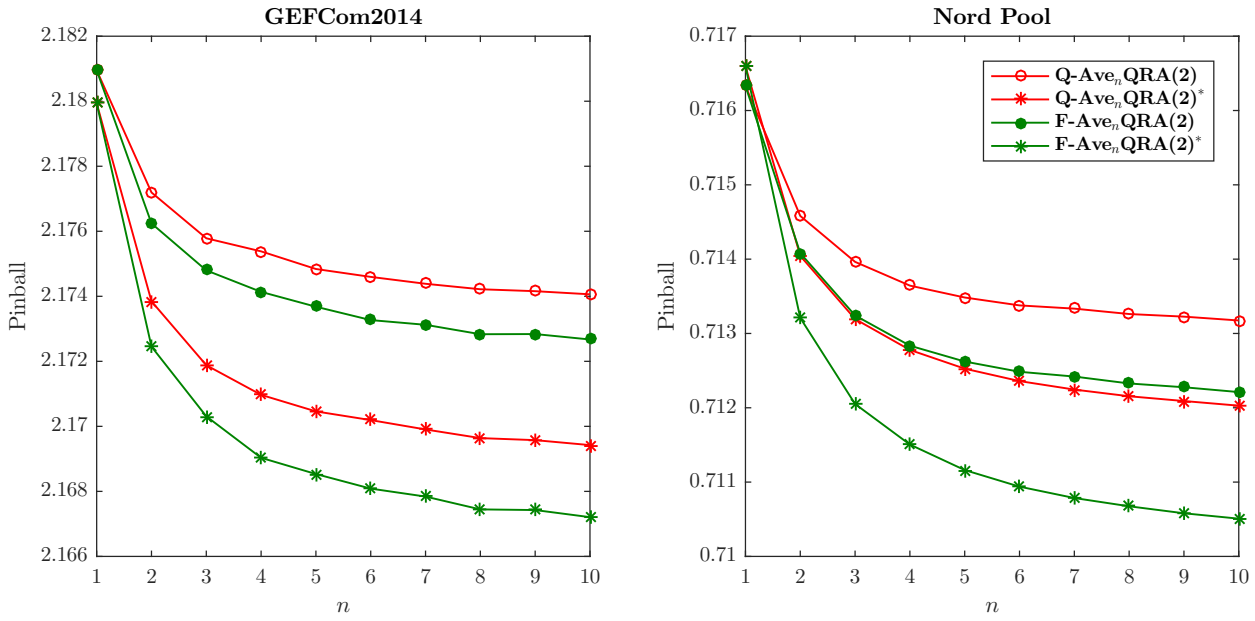


Figure 6: Comparison of **QRA(2)**- and **QRA(2)**$^*$-type combined models in terms of the pinball loss in the 'probabilistic' test period: for GEFCom2014 (*left panel*) and Nord Pool (*right panel*). To construct these plots we use 10 different probabilistic forecasts based on five (**QRA(2)**) and 20 (**QRA(2)**$^*$) point forecasts of a SCANN network. The color scheme is the same as in Fig. 5.

reasonable to consider a larger pool of point forecasts (i.e., more information), but not to increase the number of probabilistic forecasts. To address this issue we consider a variant of the **QRA**($N$) model in which each point forecast is used only once; we denote it by **QRA**($N$)$^*$. Namely, if we want to obtain $n$ probabilistic QRA-based forecasts we have to compute $n \times N$ point predictions. For instance, to obtain an **F-Ave**$_{10}$**QRA**(2)$^*$ forecast we require 20 point forecasts, whereas only 5 for **F-Ave**$_{10}$**QRA**(2), see Fig. 4. Note also, that a single run of **QRA**($N$)$^*$ will be identical to **QRA**($N$), i.e., this new concept makes a difference only in the context of combining predictive distributions.

In Figure 6 we compare **QRA**(2)- and **QRA**(2)$^*$-type combined models. Clearly, the **QRA**(2)$^*$ concept outperforms **QRA**(2) for both datasets. Note, that the computational cost for a one week forecast increases only by $0.23 \times 15 \approx 3.5$ seconds (15 additional point forecasts need to be generated). Like before, averaging probabilities (**F-Ave**) outperforms averaging quantiles (**Q-Ave**) for all considered values of $n$.

### *4.3. Diebold-Mariano (DM) tests*

The WMAE values analyzed in Section 4.1 or the pinball scores studied in Section 4.2 can be used to provide a ranking of models, but not statistically significant conclusions on differences in forecasting performance. In this Section we compute the Diebold and Mariano (1995) test (abbreviated 'DM test'), which takes into account the correlation structure of prediction errors and performs a pairwise comparison.

In the EPF literature, the DM test is usually conducted separately for each of the load periods of the day (see Nowotarski and Weron, 2018, for a review). However, here we follow Ziel and Weron

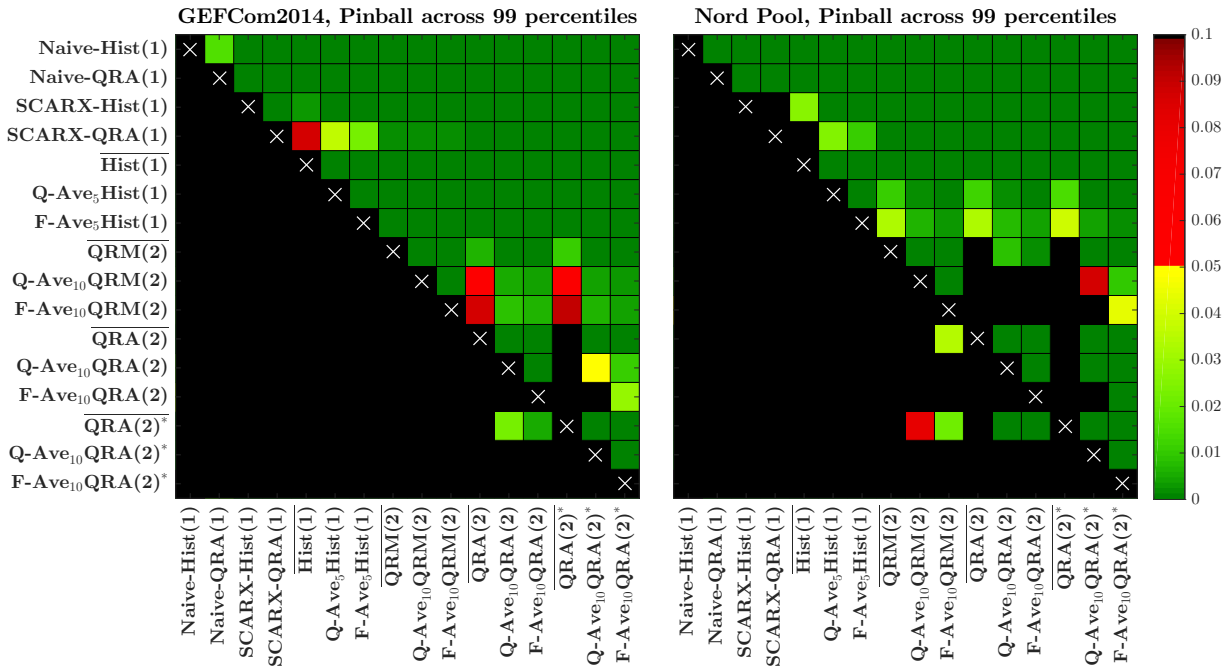


Figure 7: Results of the multivariate DM test defined by the multivariate loss differential series in Eqn. (4) for 16 selected models and the GEFCom2014 (*left panel*) and Nord Pool (*right panel*) datasets. We use a heat map to indicate the range of the $p$-values – the closer they are to zero ($\rightarrow$ dark green) the more significant is the difference between the forecasts of a model on the X-axis (better) and the forecasts of a model on the Y-axis (worse).

(2018) and conduct the *multivariate* (or *vectorized*) variant of the DM test, where only one statistic for each pair of models is computed based on the 24-dimensional vector of errors (or scores) for each day. Namely, if we denote by $\pi_{X,d} = (\pi_{X,d,1}, \ldots, \pi_{X,d,24})'$ and $\pi_{Y,d} = (\pi_{Y,d,1}, \ldots, \pi_{Y,d,24})'$ the vectors of pinball scores for day $d$ of models $X$ and $Y$, respectively, then the multivariate loss differential series in the $\|\cdot\|_1$-norm is given by:

$$\Delta_{X,Y,d} = \|\pi_{X,d}\|_1 - \|\pi_{Y,d}\|_1, \tag{4}$$

where $\|\pi_{X,d}\|_1 = \sum_{h=1}^{24} |\pi_{X,d,h}|$. For each model pair and each dataset we compute the $p$-value of two one-sided DM tests: (i) a test with the null hypothesis $H_0 : E(\Delta_{X,Y,d}) \leq 0$, i.e., the outperformance of the probabilistic forecasts of $Y$ by those of $X$, and (ii) the complementary test with the reverse null $H_0^R : E(\Delta_{X,Y,d}) \geq 0$, i.e., the outperformance of the probabilistic forecasts of $X$ by those of $Y$. As in the standard DM test, we assume that the loss differential series is covariance stationary.

In Figure 7 we plot the results for the multivariate DM-test for 16 selected models and both datasets. The models include both **Naïve** and **SCARX** benchmarks, the best (*ex-post*) models based on **Hist**($N$), **QRM**($N$), **QRA**($N$) and **QRA**($N$)$^*$, and their quantile (**Q-Ave**) and probability(**F-Ave**) averages. In both panels we see the corresponding $p$-values of the conducted pairwise comparisons: green and yellow squares indicate statistical significance at the 5% level (with the darkest green corresponding to close to zero $p$-values), red squares indicate weak significance with a $p$-value between 5% and 10%, while black denote no significance (i.e., a $p$-value of 10% or more). For instance, we see in the right panel that the first row is dark green, so that the forecasts of every

model significantly outperform those of the **Naïve-Hist** benchmark. In both panels we see that the columns which correspond to **F-Ave**$_{10}$**QRA**$(2)^*$ are green or yellow, meaning that this combination leads to significantly better forecasts than all other models. As can be seen in both panels, the model classes are ordered from the worst to the best performing (on average). Within each class, models with **F-Ave**$^*$ averaging typically significantly outperform **Q-Ave**$^*$ and the expected values, i.e., $\overline{***}$ models. Overall, the best model is **F-Ave**$_{10}$**QRA(2)**$^*$. On the other hand, all benchmarks are always significantly outperformed by models based on SCANN forecasts. The latter clearly demonstrates the usefulness of the SCANN concept, and forecast averaging in particular.

## 5. Conclusions

Conducting an extensive empirical study involving autoregressive and NARX-type neural network models and a test ground of nearly six/five years of hourly electricity prices for evaluating point/probabilistic forecasts, we have addressed three important questions:

- Does the *Seasonal Component Artificial Neural Network* (SCANN) approach bring benefits also in the probabilistic forecasting context?
- If so, can it be implemented efficiently to yield accurate predictions within a reasonable computational time?
- Given that averaging of neural network forecasts can be conducted at two levels, is it better to combine their point or probabilistic forecasts?

The answer is affirmative to the first two questions. Indeed, SCANN models estimated using the *Fast Artificial Neural Networks* (FANN) library have turned out to be extremely powerful forecasting tools, not only much more accurate but also much faster to calibrate than the Matlab-trained nets considered in Marcjasz et al. (2018). In their case, the gains from using committee machines are rather small, definitely much less pronounced than for the Matlab-trained nets. Moreover, the SCANN-implied probabilistic forecasts significantly outperform SCARX-implied predictions – consistently across two very distinct datasets and maintaining the computational time vs. forecast accuracy balance.

Regarding the third question, we find that in terms of accuracy it is better to construct probabilistic forecasts directly from point predictions, i.e., to answer the question posed in the title – we can say that it is better to combine probabilistic than point forecasts. However, if speed is a critical issue, running quantile regression on combined point forecasts (i.e., committee machines) may be an option worth considering. Finally, we confirm an observation made by Uniejewski et al. (2018) for SCARX models that averaging probabilities outperforms averaging quantiles when combining predictive distributions in EPF. This is in contrast to typical financial applications (Lichtendahl et al., 2013).

## Acknowledgments

This work was partially supported by the National Science Center (NCN, Poland) through grant no. 2015/17/B/HS4/00334. Calculations have been carried out using resources provided by

the Wrocław Center for Networking and Supercomputing (WCSS; http://wcss.pl) under grant no. 466.

## References

Afanasyev, D., Fedorova, E., 2016. The long-term trends on the electricity markets: Comparison of empirical mode and wavelet decompositions. Energy Economics 56, 432–442.

Andrawis, R., Atiya, A., El-Shishiny, H., 2011. Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition. International Journal of Forecasting 27 (3), 672–688.

Caldana, R., Fusai, G., Roncoroni, A., 2017. Electricity forward curves with thin granularity: Theory and empirical evidence in the hourly epexspot market. European Journal of Operational Research 261 (2), 715–734.

Conejo, A. J., Contreras, J., Espínola, R., Plazas, M. A., 2005. Forecasting electricity prices for a day-ahead pool-based electric energy market. International Journal of Forecasting 21, 435–462.

Diebold, F. X., Mariano, R. S., 1995. Comparing predictive accuracy. Journal of Business and Economic Statistics 13, 253–263.

Gaillard, P., Goude, Y., Nedellec, R., 2016. Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting. International Journal of Forecasting 32 (3), 1038–1050.

Gneiting, T., 2011. Quantiles as optimal point forecasts. International Journal of Forecasting 27 (2), 197–207.

Hagan, M., Demuth, H., Beale, M., De Jesús, O., 2014. Neural Network Design, 2nd ed. Martin Hagan.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9 (8), 1735–1780.

Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., Hyndman, R. J., 2016. Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond. International Journal of Forecasting 32 (3), 896–913.

Hyndman, R., Athanasopoulos, G., 2013. Forecasting: Principles and practice. Online at http://otexts.org/fpp/.

Janczura, J., Trück, S., Weron, R., Wolff, R., 2013. Identifying spikes and seasonal components in electricity spot price data: A guide to robust modeling. Energy Economics 38, 96–110.

Jones, E., Oliphant, T., Peterson, P., et al., 2001. SciPy: Open source scientific tools for Python. Http://www.scipy.org.

Keles, D., Scelle, J., Paraschiv, F., Fichtner, W., 2016. Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks. Applied Energy 162, 218–230.

Koenker, R. W., 2005. Quantile Regression. Cambridge University Press.

Kristiansen, T., 2012. Forecasting Nord Pool day-ahead prices with an autoregressive model. Energy Policy 49, 328–332.

Lichtendahl, K. C., Grushka-Cockayne, Y., Winkler, R. L., 2013. Is it better to average probabilities or quantiles? Management Science 59 (7), 1594–1611.

Lisi, F., Nan, F., 2014. Component estimation for electricity prices: Procedures and comparisons. Energy Economics 44, 143–159.

Liu, B., Nowotarski, J., Hong, T., Weron, R., 2017. Probabilistic load forecasting via Quantile Regression Averaging on sister forecasts. IEEE Transactions on Smart Grid 8, 730–737.

Maciejowska, K., Nowotarski, J., 2016. A hybrid model for GEFCom2014 probabilistic electricity price forecasting. International Journal of Forecasting 32 (3), 1051–1056.

Maciejowska, K., Nowotarski, J., Weron, R., 2016. Probabilistic forecasting of electricity spot prices using Factor Quantile Regression Averaging. International Journal of Forecasting 32 (3), 957–965.

Marcjasz, G., Uniejewski, B., Weron, R., 2018. On the importance of the long-term seasonal component in day-ahead electricity price forecasting with NARX neural networks. International Journal of Forecasting (doi: 10.1016/j.ijforecast.2017.11.009).

Misiorek, A., Trück, S., Weron, R., 2006. Point and interval forecasting of spot electricity prices: Linear vs. non-linear time series models. Studies in Nonlinear Dynamics & Econometrics 10 (3), Article 2.

Nissen, S., 2007. Large Scale Reinforcement Learning using Q-SARSA($\lambda$) and Cascading Neural Networks. MSc Thesis, Department of Computer Science, University of Copenhagen Denmark.

Nogales, F. J., Contreras, J., Conejo, A. J., Espinola, R., 2002. Forecasting next-day electricity prices by time series models. IEEE Transactions on Power Systems 17, 342–348.

Nowotarski, J., Raviv, E., Trück, S., Weron, R., 2014. An empirical comparison of alternate schemes for combining electricity spot price forecasts. Energy Economics 46, 395–412.

Nowotarski, J., Tomczyk, J., Weron, R., 2013. Robust estimation and forecasting of the long-term seasonal component of electricity spot prices. Energy Economics 39, 13–27.

Nowotarski, J., Weron, R., 2015. Computing electricity spot price prediction intervals using quantile regression and forecast averaging. Computational Statistics 30 (3), 791–803.

Nowotarski, J., Weron, R., 2016. On the importance of the long-term seasonal component in day-ahead electricity price forecasting. Energy Economics 57, 228–235.

Nowotarski, J., Weron, R., 2018. Recent advances in electricity price forecasting: A review of probabilistic forecasting. Renewable and Sustainable Energy Reviews 81, 1548–1568.

Serinaldi, F., 2011. Distributional modeling and short-term forecasting of electricity prices by Generalized Additive Models for location, scale and shape. Energy Economics 33, 1216–1226.

Shrivastava, N., Panigrahi, B., 2014. A hybrid wavelet-ELM based short term price forecasting for electricity markets. International Journal of Electrical Power and Energy Systems 55, 41–50.

Uniejewski, B., Marcjasz, G., Weron, R., 2018. On the importance of the long-term seasonal component in day-ahead electricity price forecasting. Part II – Probabilistic forecasting. Energy Economics (doi: 10.1016/j.eneco.2018.02.007).

Uniejewski, B., Nowotarski, J., Weron, R., 2016. Automated variable selection and shrinkage for day-ahead electricity price forecasting. Energies 9, 621.

Weron, R., 2006. Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach. John Wiley & Sons, Chichester.

Weron, R., 2014. Electricity price forecasting: A review of the state-of-the-art with a look into the future. International Journal of Forecasting 30 (4), 1030–1081.

Weron, R., Misiorek, A., 2008. Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models. International Journal of Forecasting 24, 744–763.

Weron, R., Zator, M., 2015. A note on using the Hodrick-Prescott filter in electricity markets. Energy Economics 48, 1–6.

Zhang, G., Qi, M., 2005. Neural network forecasting for seasonal and trend time series. European Journal of Operational Research 160 (2), 501–514.

Zhang, Y., Liu, K., Qin, L., An, X., 2016. Deterministic and probabilistic interval prediction for short-term wind power generation based on variational mode decomposition and machine learning methods. Energy Conversion and Management 112, 208–219.

Ziel, F., 2016. Forecasting electricity spot prices using LASSO: On capturing the autoregressive intraday structure. IEEE Transactions on Power Systems 31 (6), 4977–4987.

Ziel, F., Weron, R., 2018. Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks. Energy Economics 70, 396–420.

# HSC Research Report Series 2018

For a complete list please visit http://ideas.repec.org/s/wuu/wpaper.html

01 *An empirical analysis of green energy adoption among residential consumers in Poland* by Anna Kowalska-Pyzalska
02 *Efficient forecasting of electricity spot prices with expert and LASSO models* by Bartosz Uniejewski and Rafał Weron
03 *A note on averaging day-ahead electricity price forecasts across calibration windows* by Katarzyna Hubicka, Grzegorz Marcjasz and Rafał Weron
04 *Household willingness to pay for green electricity in Poland* by Anna Kowalska-Pyzalska and David Ramsey
05 *Probabilistic electricity price forecasting with NARX networks: Combine point or probabilistic forecasts?* by Grzegorz Marcjasz, Bartosz Uniejewski and Rafał Weron